OccuSeg: Occupancy-aware 3D Instance Segmentation

Lei Han[1,2], Tian Zheng[1], Lan Xu[1,2], and Lu Fang[1]✉

[1]Tsinghua University [2]Hong Kong University of Science and Technology

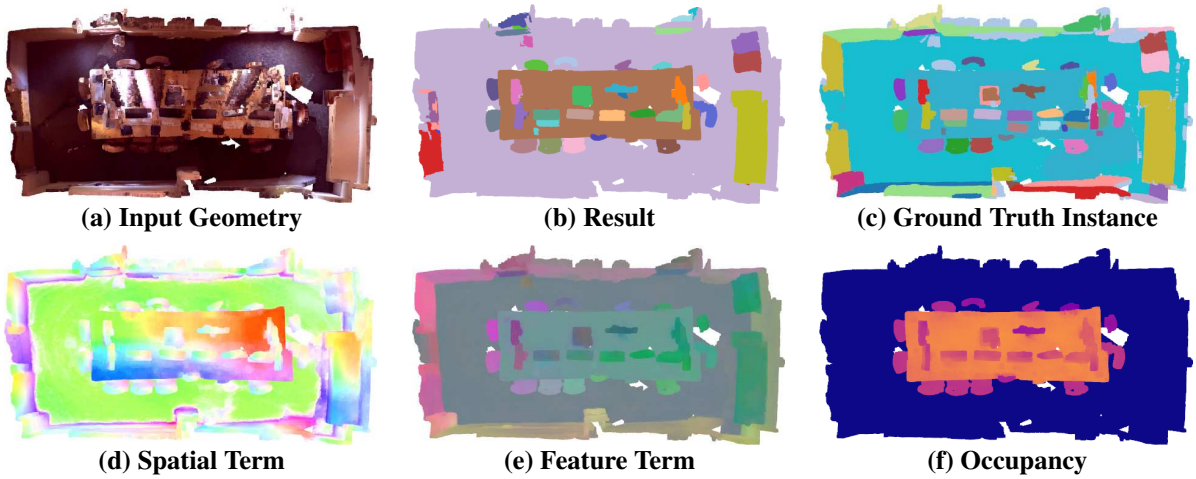

Figure 1. Given the input colored point cloud, occupancy size is regressed for each voxel, which predicts the number of voxels occupied by its belonging instance. An adaptive clustering scheme jointly considers both the occupancy information and embedding distance is further applied for 3D instance segmentation.

Abstract

3D instance segmentation, with a variety of applications in robotics and augmented reality, is in large demands these days. Unlike 2D images that are projective observations of the environment, 3D models provide metric reconstruction of the scenes without occlusion or scale ambiguity. In this paper, we define "3D occupancy size", as the number of voxels occupied by each instance. It owns advantages of robustness in prediction, on which basis, OccuSeg, an occupancy-aware 3D instance segmentation scheme is proposed. Our multi-task learning produces both occupancy signal and embedding representations, where the training of spatial and feature embedding varies with their difference in scale-aware. Our clustering scheme benefits from the reliable comparison between the predicted occupancy size and the clustered occupancy size, which encourages hard samples being correctly clustered and avoids over segmentation. The proposed approach achieves state-of-the-art performance on 3 real-world datasets, i.e. ScanNetV2, S3DIS and SceneNN, while maintaining high efficiency.

1. Introduction

The past ten years have witnessed a rapid development of real-time 3D reconstruction technologies [31, 32, 5, 45, 14] with the popularity of commercial RGB-D depth sensors like Kinect, Xtion, etc. Given the reconstructed scene, there is an increasing attention for instance-level semantic understanding of the 3D environment. More specifically, 3D instance segmentation aims to recognize points belonging to the same object and simultaneously infer their semantic class, which serves as the fundamental technique for mobile robots as well as augmented/virtual reality applications.

Although scene understanding on 2D images has achieved significant progress recently with the development of deep learning techniques, the irregularity of 3D data introduces new challenges beyond the capability of 2D solutions. As demonstrated in previous works [17], directly projecting the state-of-the-art 2D instance segmentation

✉ Corresponding author. Mail: fanglu@sz.tsinghua.edu.cn.

This work was supported in part by Natural Science Foundation of China (NSFC) under contract No. 61722209 and 6181001011, and was carried out at Tsinghua University.

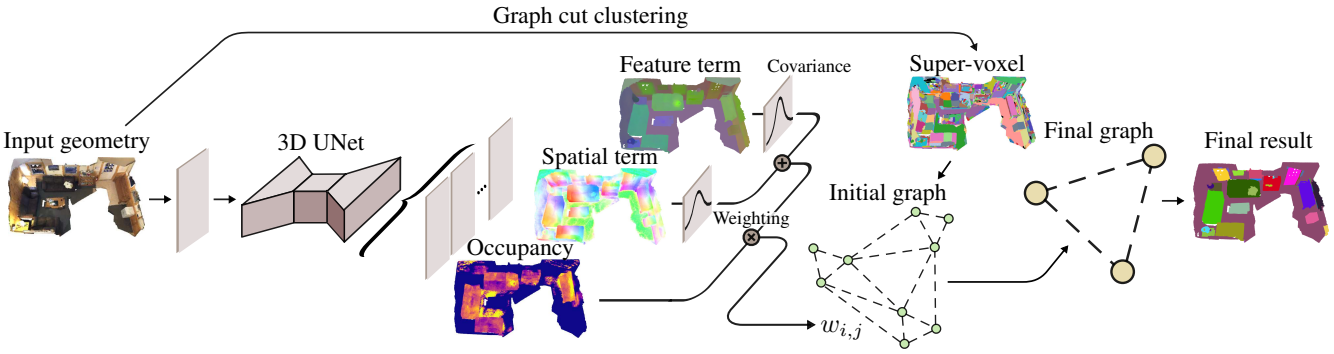

Figure 2. Overview of the proposed instance segmentation scheme. For the input point cloud, our method takes RGB feature as input and employ 3D UNet for point-wise feature learning. The learned feature is decoded to various representations though a fully connected layer for 3D instance segmentation.

MaskRCNN [16] predictions into 3D space leads to poor performance, which inspires better solutions by incorporating 3D geometry information into the network design. A popular solution for 3D instance segmentation [26, 41, 21] is to marry the powerful 3D feature extractors (spatially sparse convolutional networks [13] or PointNet++ [37]) with conventional 2D image instance segmentation techniques [16, 7, 27]. Such existing 3D solutions pay less attention to utilizing the inherent property of the 3D model itself, which provides metric reconstruction of the environment without occlusion or scale ambiguity.

In this paper, we propose an occupancy-aware 3D instance segmentation approach, OccuSeg. It takes the 3D geometry model as input, and produces point-wise predictions of instance level semantic information, as illustrated in Fig. 1. Given the insight that the 3D metric space provides more reliable perception than the 2D image-based projective observations for the 3D scene, we particularly introduce "3D occupancy signal", representing *the number of voxels occupied by each instance*. Such an occupancy signal represents the inherent and fundamental property of each 3D instance, showing a strong potential to handle the ambiguity of scale, location, texture, lighting and occlusion under the 3D setting. Thus, we encode the novel occupancy signal into the conventional 3D instance segmentation pipeline, i.e., *learning* stage followed by *clustering* stage. In our occupancy-aware approach, both the learning and clustering stages fully utilize the characteristic of the occupancy signal, leading to competitive performance on public datasets. The considerable gain on mAP (around 12.3 in mAP) further demonstrates that our occupancy-aware approach owns the superiority of preserving the inherent and fundamental nature of the instances in the 3D environment.

More specifically, the *learning* stage takes a colored 3D scene as input, and utilizes the spatially sparse convolution approaches [12] to extract a hybrid vector for each voxel [26, 25, 21]. It not only learns the classic embedding such as spatial (Fig. 1(d)) and feature embedding (Fig. 1(e)), but also produces an occupancy signal (Fig. 1(f)) that implies the object-level volume. To make full use of both the semantic and geometric information, our feature and spatial embedding are explicitly supervised with different objectives, and are further combined through covariance estimation for both feature and spatial embedding distance. For the *clustering* stage, the 3D input point cloud is grouped into super-voxels based on the geometric and appearance constraints using a graph-based segmentation algorithm [9]. Then, to merge the super-voxels with similar feature embedding into the same instance, we utilize an adaptive threshold to evaluate the similarity between the embedding distance and the occupancy size. Aided by the reliable comparison between the predicted occupancy size and the clustered occupancy size, our clustering encourages hard samples to be correctly clustered and eliminates the false positives where partial instances are recognized as an independent instance. The technical contributions are summarized as follows.

- We present an occupancy-aware 3D instance segmentation scheme OccuSeg. It achieves state-of-the-art performance on three public datasets: ScanNetV2 [4], S3DIS [1] and SceneNN [18], ranking first in all metrics with a significant margin while remaining high efficiency, e.g., 12.3 gain in mAP on the ScanNetV2 benchmark.
- In particular, a novel occupancy signal is proposed in this paper, which predicts the number of occupied voxels for each instance. The occupancy signal is learnt jointly with a combination of feature and spatial embedding and employed to guide the clustering stage of 3D instance segmentation.

2. Related Work

2D Instance Segmentation. 2D Instance segmentation methods are generally divided into two categories: proposal-based and proposal-free approaches. Proposal-

based methods [10, 6, 16, 15, 23, 44] firstly generate region proposals (predefined rectangles) that contain objects and further classify pixels inside each proposal as objects or background. By arguing that convolutional operators are translational invariant and thus cannot distinguish similar objects at different places well, Novotny *et al*. [33] propose semi-convolutional operators based on the coordinates of each pixel for better instance segmentation.

On the other hand, proposal-free methods [24, 7, 8, 20, 22] learn an embedding vector for each pixel and apply a clustering step in the embedding space as post processing for instance segmentation. Brabandere *et al*. [7] propose to train a per-pixel embedding vector and adopt a discriminative cost to encourage pixels belonging to the same instance to be as close as possible, while the embedding center of different instances to be far away from each other. Liang *et al*. [24] regress an offset vector pointing to the object center for each pixel and further use the predicted centers for instance segmentation from a "voting" perspective [22]. Recently, Neven *et al*. [30] introduce a learnable clustering bandwidth instead of learning embedding using hand-crafted cost functions, achieving accurate instance segmentation in real-time.

While all these approaches have achieved promising results in the 2D domain, the extension to the 3D domain is non-trivial. How to utilize the fundamental property of 3D instance remains a challenging problem.

3D Instance Segmentation. Unlike 2D images with regular pixel grids, the irregular distribution of 3D point clouds in physical space raises new challenges for 3D instance segmentation. Pioneer works [43, 40, 17] have tried to directly extend 2D convolutional neural networks to 3D space by voxelizing the input points into uniform voxels, and applying 3D convolutions instead. Yet most computations are wasted on the inactive empty voxels. Thus, recent methods utilize more feasible 3D feature extractors to tackle this problem. Point-based instance segmentation approaches [41, 46, 48] directly consume unordered point clouds as input and use a permutation invariant neural network PointNet [36, 37] for feature extraction. While volumetric approaches [26, 25, 21] employ Spatially-Sparse Convolutional Networks (SSCN) [12, 2] to omit computations on inactive voxels using the sparse convolution technique.

Specifically, SGPN [41] proposes to learn a similarity matrix for all point pairs, based on which, similar points are merged for instance segmentation. 3D BoNet [46] directly predicts the bounding boxes of objects for efficient instance segmentation. GSPN [48] introduces a generative shape proposal network and relies on object proposals to identify instances in 3D point clouds. VoteNet [35] predicts offset vectors to the corresponding object centers for seed points, followed by a clustering module to generate object proposals. Additionally, 3DSIS [17] jointly learns 2D and 3D features by back projecting features extracted from 2D convolution on images to 3D space. It further applies 3D convolution for volumetric feature learning for proposal-based 3D instance segmentation. For proposal-free 3D instance segmentation, MASC [26] combines the SSCN architecture with instance affinity prediction across multiple scales. Liang *et al.* [25] apply embedding learning [7] on top of the superior performance of SSCN. Lahoud *et al.* [21] further combine directional information of each object with semantic feature embedding.

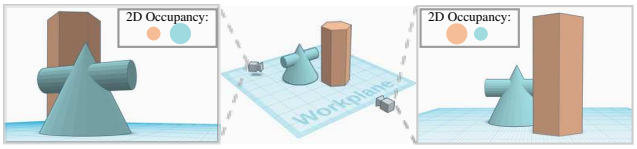

Figure 3. Toy example of 2D observations at different view angles of the same 3D scene. The number of occupied pixels/voxels of each instance (denoted as occupancy) is uncertain on 2D image, yet can be predicted robustly for the reconstructed 3D model.

3. Methods

Recall our goal is that, we take a voxelized 3D colored scene as input , and produce a 3D object instance label for each voxel, where the voxels belonging to the same object share an unique instance label.

Examining the aforementioned approaches, few of them explicitly utilize the inherent nature of 3D models that differs from 2D image observations: reconstruction of the environment in metric space without occlusion or scale ambiguity. As shown in Fig. 3, for the same instance in 3D space, its observations on 2D images can vary greatly. The number of occupied pixels/voxels of each instance (denoted as occupancy) is unpredictable on 2D image, yet can be predicted robustly from the reconstructed 3D model.

On the basis of occupancy signal, we propose an occupancy-aware 3D instance segmentation scheme. The pipeline is illustrated in Fig. 2. While it follows the classic learning followed by clustering procedure, both the learning stage and clustering stage differ from existing approaches. First, the input 3D scene is voxelized at a resolution of $2cm$ and is then fed into a 3D convolutional neural network (U-Net [38]) for feature extraction. Then, the learned feature is forwarded to task-specific heads to learn different representations for each input voxel, including semantic segmentation, which aims to assign a class label, feature and spatial embedding, as well as occupancy regression (Sec. 3.1). Finally, a graph-based occupancy-aware clustering scheme is performed, which utilizes both the predicted occupancy information and the feature embedding from the previous stage (Sec. 3.2). Note that all the 3D convolutions are realized using a submanifold sparse convolutional network [13]

to employ the sparsity nature of input 3D scene. The details of the network are provided in the Appendix.

3.1. Multi-task Learning

In order to jointly leverage the inherent occupancy and semantic and spatial information from the 3D scene, we propose a multi-task learning framework to learn task-specific representations for the i-th input voxel, including (1) $\mathbf{c}_i$ for the semantic segmentation, which aims to assign a class label; (2) $\mathbf{s}_i$ and $\mathbf{d}_i$ for the joint feature and spatial embedding, as well as the corresponding $\mathbf{b}_i$ for covariance prediction to fuse feature and spatial information; and (3) o_i for the occupancy regression. The network is trained to minimize a joint cost function $\mathcal{L}_{\text{joint}}$:

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{c}} + \mathcal{L}_{\text{e}} + \mathcal{L}_{\text{o}}. \tag{1}$$

Here $\mathcal{L}_{\text{c}}$ is a conventional cross-entropy loss [11] for semantic segmentation. $\mathcal{L}_{\text{e}}$ aims to learn an embedding vector that considers jointly feature and spatial embedding for instance segmentation (Sec. 3.1.1). $\mathcal{L}_{\text{o}}$ serves for the regression of the occupancy size of each voxel's belonging instance (Sec. 3.1.2).

3.1.1 Embedding Learning

Unlike previous methods [33] that concatenate the feature and spatial embedding directly, we propose to separate them explicitly and supervise their learning process with different objectives. Our key observation is that while spatial embedding is scale-aware and has an explicit physical explanation, such as an offset vector from current voxel to the spatial center of its belonging instance, feature embedding suffers from inherently ambiguous scale, and thus has to be regularized using additional cost functions. Both embeddings are further regularized using the covariance estimation. Our learning function for embedding $\mathcal{L}_{\text{e}}$ consists of three terms, i.e., spatial term $\mathcal{L}_{\text{sp}}$, feature term $\mathcal{L}_{\text{se}}$, and covariance term $\mathcal{L}_{\text{cov}}$,

$$\mathcal{L}_{\text{e}} = \mathcal{L}_{\text{sp}} + \mathcal{L}_{\text{se}} + \mathcal{L}_{\text{cov}}. \tag{2}$$

Spatial Term. Spatial embedding $\mathbf{d}_i$ for the i-th voxel is a 3-dimensional vector that regresses to the object center, which is supervised using the following spatial term:

$$\mathcal{L}_{\text{sp}} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_c}||\mathbf{d}_i + \mu_i - \frac{1}{N_c}\sum_{i=1}^{N_c}\mu_i||, \tag{3}$$

where C is the number of instances in the input 3D scene, N_c is the number of voxels in the c-th instance, and μ_i represents the 3D position of the i-th voxel of the c-th instance.

Feature Term. Feature embedding $\mathbf{s}_i$ is learned using a discriminative loss function [7] that consists of three terms:

$$\mathcal{L}_{\text{se}} = \mathcal{L}_{\text{var}} + \mathcal{L}_{\text{dist}} + \mathcal{L}_{\text{reg}}, \tag{4}$$

where the variance term $\mathcal{L}_{\text{var}}$ draws current embedding towards the mean embedding of each instance, the distance term $\mathcal{L}_{\text{dist}}$ pushes instances away from each other, and the regularization term $\mathcal{L}_{\text{reg}}$ draws all instances towards the origin to keep the activation bounded. The detailed formulations are as follows.

$$\mathcal{L}_{\text{var}} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_C}[||\mathbf{u}_c - \mathbf{s}_i|| - \delta_v]_+^2, \tag{5}$$

$$\mathcal{L}_{\text{dist}} = \frac{1}{C(C-1)}\sum_{c_A=1}^{C}\sum_{c_B=c_A+1}^{C}[2\delta_d - ||\mathbf{u}_{c_A} - \mathbf{u}_{c_B}||]_+^2, \tag{6}$$

$$\mathcal{L}_{\text{reg}} = \frac{1}{C}\sum_{c=1}^{C}||\mathbf{u}_c||. \tag{7}$$

Here, $\mathbf{u}_c = \frac{1}{N_c}\sum_{i=1}^{N_c}\mathbf{s}_i$ represents the mean feature embedding of the c-th instance. The predefined thresholds δ_v and δ_d are set to be 0.1 and 1.5, ensuring that the intra-instance embedding distance is smaller than the inter-instance distance.

Covariance Term. The covariance term aims to learn an optimal clustering region for each instance. Let $\mathbf{b}_i = (\sigma_s^i, \sigma_d^i)$ denote the predicted feature/spatial covariance for the i-th voxel in the c-th instance. By averaging $\mathbf{b}_i$, we obtain (σ_s^c, σ_d^c), the embedding covariance of the c-th instance. Then, the probability of the i-th voxel belonging to the c-th instance, denoted as p_i, is formulated as:

$$p_i = \exp(-(\frac{||\mathbf{s}_i - \mathbf{u}_c||}{\sigma_s^c})^2 - (\frac{||\mu_i + \mathbf{d}_i - \mathbf{e}_c||}{\sigma_d^c})^2), \tag{8}$$

where $\mathbf{e}_c = \frac{1}{N_c}\sum_{k=0}^{N_c}(\mu_k + \mathbf{d}_k)$ represents the predicted spatial center of the c-th instance. Since p_i is expected to be larger than 0.5 for voxels that belong to the c-th instance, the covariance term is then formulated by a binary cross-entropy loss,

$$\mathcal{L}_{\text{cov}} = -\frac{1}{C}\sum_{c=1}^{C}\frac{1}{N}\sum_{i=1}^{N}[y_i log(p_i) + (1-y_i)log(1-p_i)], \tag{9}$$

where $y_i = 1$ indicates i belongs to c and $y_i = 0$ otherwise, N indicates the number of points in the input point cloud.

3.1.2 Occupancy Regression

To utilize the occupancy information under the 3D setting, for the i-th voxel in the c-th instance, we predict a positive value o_i to indicate the number of voxels occupied by the current instance. Then, the average of o_i will serve as the predicted occupancy size of the current instance. For more robust prediction, we regress the logarithm instead of the

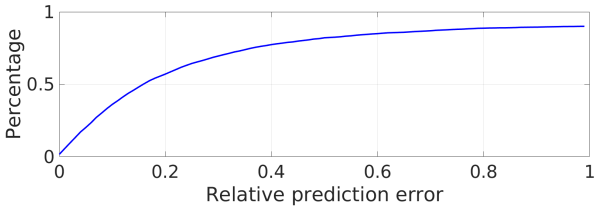

Figure 4. Cumulative distribution function of the relative prediction error on the validation set of the ScanNetV2 [4].

original value and formulate the following occupancy term,

$$\mathcal{L}_{\mathrm{o}} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_c}||o_i - \log(N_c)||, \tag{10}$$

where N_c is the number of voxels in the c-th instance.

To evaluate the feasibility of our occupancy prediction strategy, we use the relative prediction error R_c to measure the occupancy prediction performance of the c-th instance,

$$R_c = \frac{|N_c - \exp(\frac{1}{N_c}\sum_{i=1}^{N_c} o_i)|}{N_c}. \tag{11}$$

We particularly plot the cumulative distribution function of R_c in Fig. 4. For over 4000 instances in the validation set of ScanNetV2 dataset [4], more than 68% instances are predicted, with a relative error smaller than 0.3, which illustrates the effectiveness of our occupancy regression for the following clustering stage.

3.2. Instance Clustering

In this subsection, based on the multi-representation learning from the previous stage, a graph-based occupancy-aware clustering scheme is introduced to tackle the 3D instance segmentation problem during inference. Specifically, we adopt a bottom-up strategy and group the input voxels into super-voxels using an efficient graph-based segmentation scheme [9]. Compared with super-pixel representations in 2D space [39, 47], super-voxel representation works better to separate different instances where the instance boundaries in 3D space is easier to identify thanks to the geometry continuity or local convexity constraints [3].

Let Ω_i denote the collection of all the voxels belonging to the super-voxel v_i, we define the spatial embedding $\mathbf{D}_i$ of v_i as,

$$\mathbf{D}_i = \frac{1}{|\Omega_i|}\sum_{k\in\Omega_i}(\mathbf{d}_i + \mu_i), \tag{12}$$

where $|\Omega_i|$ represents the number of voxels in Ω_i. The feature embedding $\mathbf{S}_i$, occupancy O_i and covariance σ_s^i, σ_d^i of v_i are computed based on a similar averaging operation for all the voxels belonging to v_i. We further define the following occupancy ratio r_i to guide the clustering step,

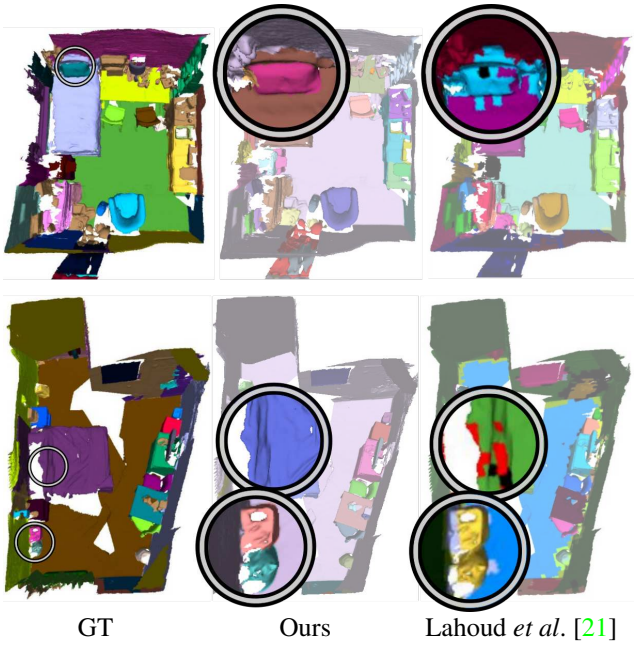

Figure 5. Qualitative comparisons between OccuSeg and a previous approach [21] on the validation set of the ScanNetV2 [4]. OccuSeg generates more consistent instance labels and successfully distinguishes nearby small instances thanks to the proposed occupancy aware clustering scheme.

$$r_i = \frac{O_i}{|\Omega_i|}. \tag{13}$$

Note that $r_i > 1$ indicates that there are too many voxels in v_i for instance segmentation, otherwise v_i should attract more voxels.

Given the super-voxel representation, an undirected graph $G = (V, E, W)$ is established, where the vertices $v_i \in V$ represent the generated super-voxels, $e_{i,j} = (v_i, v_j) \in E$ indicates the pairs of vertices with a weight $w_{i,j} \in W$. The weight $w_{i,j}$ represents the similarity between v_i and v_j. Here $w_{i,j}$ is formulated as

$$w_{i,j} = \frac{\exp(-(\frac{||\mathbf{S_i}-\mathbf{S_j}||}{\sigma_s})^2 - (\frac{||\mathbf{D_i}-\mathbf{D_j}||}{\sigma_d})^2)}{max(r, 0.5)}, \tag{14}$$

where σ_s, σ_d and r represent the feature covariance, spatial covariance and occupancy ratio of the virtual super-voxel that merges both v_i and v_j.

Note that a larger weight indicates a higher possibility that v_i and v_j belong to the same instance. And during the calculation of the merging weight, our occupancy ratio helps to punish over-segmented instances and encourages partial instances to be merged together as shown in Fig. 5.

For all the edges in E, we select edge $e_{i,j}$ with the highest weight $w_{i,j}$ and merge v_i, v_j as a new vertex if

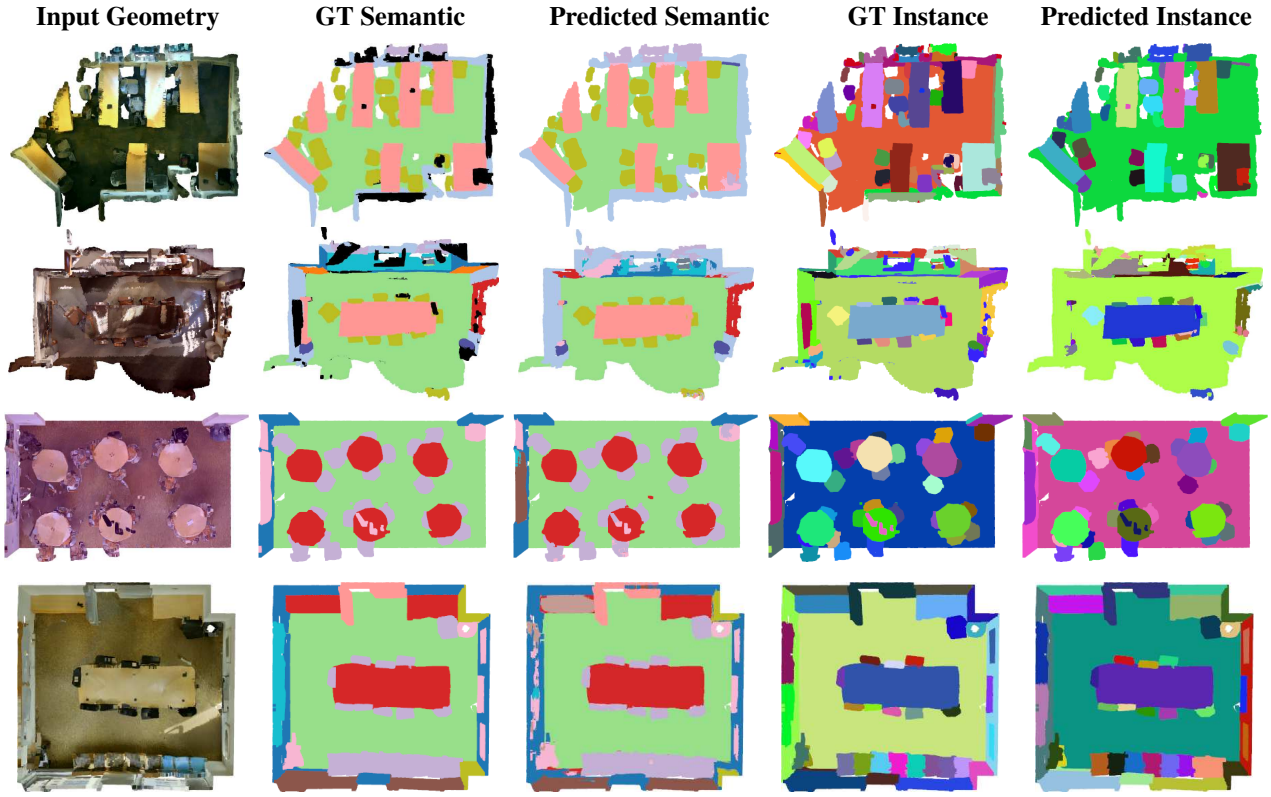

Figure 6. Representative 3D instance segmentation results on the validation set of public datasets, including ScanNetV2 [4] and S3DIS [1].

$w_{i,j} > T_0$, where the merge threshold T_0 is set to be 0.5. The graph G is then updated after every merge operation. This process is iterated until none of the weight is larger than T_0. Finally, the remaining vertices in G are labeled as instances if their occupancy ratio r satisfies the constraint of $0.3 < r < 2$ to reject false positives in instance segmentation.

3.3. Network Training

We employ a simple UNet-like structure [38] for feature extraction from the input point cloud with color information. Network details are presented in the Appendix. For the sake of efficiency, the chunk-based sparse convolution strategy in [19] is adopted, which is $4\times$ faster than the original implementation of the SCN [13]. The network is trained using Adam optimizer with an initial learning rate of 1e-3. For all the datasets including ScanNetV2 [4], Stanford3D [1] and SceneNN [18] as shown in the experiments of Sec. 4, we use the same hyper-parameters and train the network from scratch for 320 epochs.

4. Experiments

In this section, we evaluate our method on a variety of challenging scenarios. For experiments on public datasets, we run our method on a PC with a NVIDIA TITAN Xp GPU and an Intel(R) Xeon(R) E5-2650 CPU. For real-world experiments, our method is conducted on the laptop Microsoft Surface Book 2 with a NVIDIA GTX 1060 (Mobile) GPU and an Intel Core i7-8650U CPU. Using the real-time 3D reconstruction method FlashFusion [14] for the 3D geometric input, we present the demo of online 3D instance segmentation on the portable device. More details are provided in the supplementary video.

We employ the popular 3D instance segmentation benchmark ScanNetV2 [4], as well as the widely used S3DIS [1] and SceneNN [18] datasets. ScanNetV2 benchmark [4] contains 1513 indoor RGBD scans with 3D instance annotations, while Stanford Large-Scale 3D Indoor Space Dataset (S3DIS) [1] contains 6 large-scale indoor areas covering over $6000m^2$ with 13 object classes. SceneNN [18] is a smaller indoor 3D dataset with 50 scans as the training set and 26 scans for evaluation, which is used to evaluate our performance under less training data.

4.1. Qualitative Evaluation

The representative 3D instance segmentation results on the validation set of public datasets are presented in Fig. 6, which demonstrate that the proposed approach achieves robust instance segmentation results for complex environments.

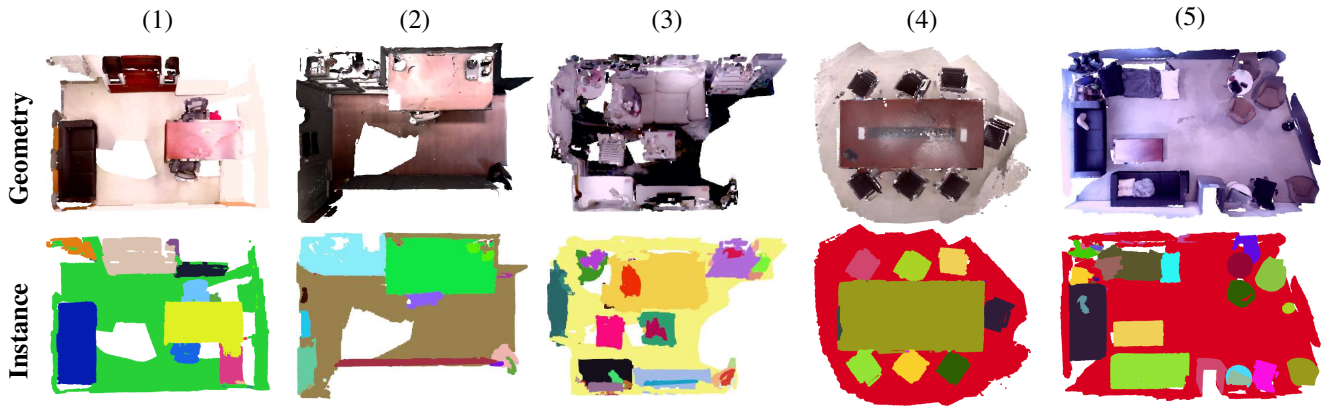

Figure 7. 3D instance segmentation results on real-world scenes. Here the 3D geometric models are reconstructed using FlashFusion [14] system, with the input being the depth and color sequences from consumer-level RGB-D camera. Our scheme generates robust instance segmentation results in real-world environments using the network trained on a public dataset, ScanNetV2 [4].

Method	**mAP**	bath	bed	bkshf	cab	chair	cntr	curt	desk	door	ofurn	pic	fridg	showr	sink	sofa	tabl	toil	wind
3D-SIS [17]	16.1	40.7	15.5	6.8	4.3	34.6	0.1	13.4	0.5	8.8	10.6	3.7	13.5	32.1	2.8	33.9	11.6	46.6	9.3
PanopticFusion [29]	21.4	25.0	33.0	27.5	10.3	22.8	0.0	34.5	2.4	8.8	20.3	18.6	16.7	36.7	12.5	22.1	11.2	66.6	16.2
3D-BoNet [46]	25.3	51.9	32.4	25.1	13.7	34.5	3.1	41.9	6.9	16.2	13.1	5.2	20.2	33.8	14.7	30.1	30.3	65.1	17.8
MTML [21]	28.2	57.7	38.0	18.2	10.7	43.0	0.1	**42.2**	5.7	17.9	16.2	7.0	22.9	51.1	16.1	49.1	31.3	65.0	16.2
Occipital-SCS	32.0	67.9	35.2	33.4	22.9	43.6	2.5	41.2	5.8	16.1	24.0	8.5	26.2	49.6	18.7	46.7	32.8	77.5	23.1
OccuSeg	**44.3**	**85.2**	**56.0**	**38.0**	**24.9**	**67.9**	**9.7**	34.5	**18.6**	**29.8**	**33.9**	**23.1**	**41.3**	**80.7**	**34.5**	**50.6**	**42.4**	**97.2**	**29.1**

Table 1. Quantitative comparison on the ScanNetV2 [4] benchmark in terms of mAP score on 18 classes. Our approach achieves **the best performance in 17 out of 18 classes**. Note that the ScanNetV2 benchmark data is accessed on 11/14/2019.

	mAP	mAP@0.5	mAP@0.25
3D-SIS [17]	16.1	38.2	55.8
3D-BoNet [46]	25.3	48.8	68.7
MASC [26]	25.4	44.7	61.5
MTML [21]	28.2	54.9	73.1
Occipital-SCS	32.0	51.2	68.8
OccuSeg	**44.3**	**63.4**	**73.9**

Table 2. Quantitative results on the ScanNetV2 [4] benchmark in terms of mAP, mAP@0.5 and mAP@0.25, respectively. Our approach outperforms previous methods by a significant margin. ScanNetV2 benchmark data is accessed on 11/14/2019.

To further verify the robustness of our method on real-world scenes, we implement our method on the basis of real-time 3D reconstruction method FlashFusion [14] for online 3D instance segmentation. As shown in Fig. 7, our network pre-trained on ScanNetV2 can generate 3D instance segmentation results robustly in real-world scenarios. More live results are provided in the supplementary video.

4.2. Quantitative Evaluation

Based on the public datasets, our methods are quantitatively compared with a number of representative existing methods, including SGPN [41], 3D-SIS [17], PanopticFusion [29], 3D-BoNet [46], MTML [21], ASIS [42] and JSIS3D [34].

	mPrec	mRec
PartNet [28]	56.4	43.4
ASIS [42]	63.6	47.5
3D-BoNet [46]	65.6	47.6
OccuSeg	**72.8**	**60.3**

Table 3. Comparison on the S3DIS [1] dataset. Our method outperforms previous methods in terms of mean Precision (mPrec) and mean recall (mRec) with an IoU threshold of 0.5.

ScanNetV2. We follow the benchmark [4] to use the mean average precision at overlap 0.25 (mAP@0.25), overlap 0.5 (mAP@0.5) and overlaps in the range $[0.5 : 0.95 : 0.05]$ (mAP) as evaluation metrics. Tab. 1 and Tab. 2 summarize the per-class mAP and the overall performance, respectively. Overall, our method achieves a significant margin on all the three metrics, especially the hardest mAP metric, indicating the effectiveness of our method for 3D instance segmentation.

S3DIS. Following the previous methods [46, 42], we employ the 6-fold cross validation and use the mean precision

Method	**mAP@0.5**	wall	floor	cabinet	bed	chair	sofa	table	desk	tv	prop
MT-PNet [34]	8.5	13.1	27.3	0.0	15.0	21.2	0.0	0.7	0.0	6.0	2.0
MLS-CRF [34]	12.1	13.9	44.5	0.0	32.9	12.9	0.0	5.7	10.8	0.0	0.8
OccuSeg	**47.1**	**39.0**	**93.8**	**5.7**	**66.7**	**91.3**	**8.7**	**50.0**	**31.6**	**76.9**	**7.14**

Table 4. Quantitative results on the SceneNN [18] dataset in terms of mAP@0.5 score of each class. Our approach achieves the best performance for all the 10 classes.

	Details	Total
SGPN [41]	network(GPU): 650 group merging(CPU): 46562 block merging(CPU): 2221	49433
ASIS [42]	network(GPU): 650 mean shift(CPU): 53886 block merging(CPU): 2221	56757
GSPN [48]	network(GPU): 500 point sampling(GPU): 2995 neighbour search(CPU): 468	3963
3D-SIS [17]	network (GPU+CPU): 38841	38841
3D-BoNet [46]	network(GPU): 650 SCN (GPU parallel): 208 block merging(CPU): 2221	2871
OccuSeg	network(GPU): 59 supervoxel(CPU): 375 clustering(GPU+CPU): 160	**594**

Table 5. The processing time (seconds) on the validation set of ScanNetV2 [4]. Note that all the other methods are evaluated based on their released codes according to [46].

	mAP	mAP@0.5	mAP@0.25
w/o_feature	36.7	51.8	62.6
w/o_spatial	42.8	58.5	69.7
w/o_occupancy	40.9	55.7	67.4
OccuSeg	44.2	60.7	71.9

Table 6. Ablation study of each component of our method on the ScanNetV2 validation split, in terms of mAP, mAP@0.5 and mAP@0.25.

(mPrec) / mean recall (mRec) with an IoU threshold 0.5 to evaluate our method in the S3DIS dataset. As shown in Tab. 3, our scheme outperforms all the previous methods by a significant margin in terms of both mPrec and mRec, indicating its ability to segment more instances precisely.

SceneNN. Similar to previous work [34], mAP@0.5 metric is adopted to evaluate our approach in the SceneNN dataset. As presented in Tab. 4, even using only 50 scans for training, our approach outperforms the previous method [34] by a significant margin (35 for mAP@0.5), which illustrates the effectiveness of our approach under small datasets.

4.3. Complexity Analysis

Maintaining high efficiency plays a vital role when applying 3D instance segmentation to mixed reality or robotics applications. Similar to the evaluation in [46], we made a comparison of the processing times on the 312 scans of indoor environments from the validation split of ScanNetV2. Both the proposal-based methods (3DSIS [17], GSPN [48] and 3D-BoNet [46]) and the proposal-free methods (SGPN [41] ASIS [42]) are concerned. The processing time of the full pipeline and the main stages are respectively reported in Tab. 5. Remarkably, our method is more than $4\times$ faster than the existed most efficient approach 3D-BoNet [46].

4.4. Ablation Study

Here, we evaluate the individual components of our method on the ScanNetV2 validation split. Let *w/o_feature* and *w/o_spatial* denote the variations of our method without the feature embedding or spatial feature embedding, respectively. To evaluate the influence of the novel occupancy signal, we disable the occupancy prediction in the learning stage and set the occupancy ratio $r = 1$ for all vertices in Eqn. 14 during the clustering stage, denoted as *w/o_occupancy*.

The quantitative comparison results of all the variations of our method are provided in Tab. 6, which demonstrate that the proposed occupancy aware scheme helps to improve the overall quality of 3D instance segmentation.

5. Discussion and Conclusion

We presented OccuSeg, an occupancy-aware instance segmentation method for 3D scenes. Our learning stage leverages feature embedding and spatial embedding, as well as a novel 3D occupancy signal to imply the inherent property of 3D objects. The occupancy signal further guides our graph-based clustering stage to correctly merge hard samples and prohibit over-segmented clusters. Extensive experimental results demonstrate the effectiveness of our method, which outperforms previous methods by a significant margin and retains high efficiency. In the future work, we will improve our method by incorporating tailored designs for partially reconstructed objects. Also, we intend to investigate the sub-object level 3D instance segmentation and further improve the efficiency, enabling the practical usage of high quality 3D instance segmentation for tremendous applications in AR/VR, gaming and mobile robots.

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543, 2016. 2, 6, 7

[2] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. *arXiv preprint arXiv:1904.08755*, 2019. 3

[3] Simon Christoph Stein, Markus Schoeler, Jeremie Papon, and Florentin Worgotter. Object partitioning using local convexity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 304–311, 2014. 5

[4] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 2, 5, 6, 7, 8

[5] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (TOG)*, 36(3):24, 2017. 1

[6] Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 534–549, Cham, 2016. Springer International Publishing. 3

[7] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017. 2, 3, 4

[8] Alireza Fathi, Zbigniew Wojna, Vivek Rathod, Peng Wang, Hyun Oh Song, Sergio Guadarrama, and Kevin P. Murphy. Semantic instance segmentation via deep metric learning. *CoRR*, abs/1703.10277, 2017. 3

[9] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004. 2, 5

[10] Ross Girshick. Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 3

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. 4

[12] Benjamin Graham. Spatially-sparse convolutional neural networks. *arXiv preprint arXiv:1409.6070*, 2014. 2, 3

[13] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9224–9232, 2018. 2, 3, 6

[14] Lei Han and Lu Fang. Flashfusion: Real-time globally consistent dense 3d reconstruction using cpu computing. In *Robotics: Science and Systems*, 2018. 1, 6, 7

[15] Zeeshan Hayder, Xuming He, and Mathieu Salzmann. Boundary-aware instance segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2, 3

[17] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4421–4430, 2019. 1, 3, 7, 8

[18] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. Scenenn: A scene meshes dataset with annotations. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 92–101. IEEE, 2016. 2, 6, 8

[19] Anonymous (Attached in the supplementary material). Realtime semantic 3d perception for immersive augmented reality. In *conditionally accepted by IEEE VR/TVCG*. IEEE, 2019. 6

[20] Shu Kong and Charless C. Fowlkes. Recurrent pixel embedding for instance grouping. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[21] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R Oswald. 3d instance segmentation via multi-task metric learning. *arXiv preprint arXiv:1906.08650*, 2019. 2, 3, 5, 7

[22] Bastian Leibe, Aleš Leonardis, and Bernt Schiele. Robust object detection with interleaved categorization and segmentation. *International journal of computer vision*, 77(1-3):259–289, 2008. 3

[23] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[24] Xiaodan Liang, Liang Lin, Yunchao Wei, Xiaohui Shen, Jianchao Yang, and Shuicheng Yan. Proposal-free network for instance-level object segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2978–2991, 2017. 3

[25] Zhidong Liang, Ming Yang, and Chunxiang Wang. 3d graph embedding learning with a structure-aware loss function for point cloud semantic instance segmentation. *arXiv preprint arXiv:1902.05247*, 2019. 2, 3

[26] Chen Liu and Yasutaka Furukawa. Masc: Multi-scale affinity with sparse convolution for 3d instance segmentation. *arXiv preprint arXiv:1902.04478*, 2019. 2, 3, 7

[27] Yiding Liu, Siyu Yang, Bin Li, Wengang Zhou, Jizheng Xu, Houqiang Li, and Yan Lu. Affinity derivation and graph merge for instance segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 686–703, 2018. 2

[28] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2019. 7

[29] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. Panopticfusion: Online volumetric semantic map-

ping at the level of stuff and things. *arXiv preprint arXiv:1903.01177*, 2019. 7

[30] Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8837–8845, 2019. 3

[31] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, 2011. 1

[32] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 32(6):169, 2013. 1

[33] David Novotny, Samuel Albanie, Diane Larlus, and Andrea Vedaldi. Semi-convolutional operators for instance segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 86–102, 2018. 3, 4

[34] Quang-Hieu Pham, Thanh Nguyen, Binh-Son Hua, Gemma Roig, and Sai-Kit Yeung. Jsis3d: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2019. 7, 8

[35] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. *arXiv preprint arXiv:1904.09664*, 2019. 3

[36] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 3

[37] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 2, 3

[38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3, 6

[39] Mathijs Schuurmans, Maxim Berman, and Matthew B Blaschko. Efficient semantic image segmentation with superpixel pooling. *arXiv preprint arXiv:1806.02705*, 2018. 5

[40] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *2017 International Conference on 3D Vision (3DV)*, pages 537–547. IEEE, 2017. 3

[41] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2569–2578, 2018. 2, 3, 7, 8

[42] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4096–4105, 2019. 7, 8

[43] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 3

[44] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8818–8826, 2019. 3

[45] L. Xu, Z. Su, L. Han, T. Yu, Y. Liu, and L. FANG. Unstructuredfusion: Realtime 4d geometry and texture reconstruction using commercialrgbd cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019. 1

[46] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. *arXiv preprint arXiv:1906.01140*, 2019. 3, 7, 8

[47] Shichao Yang, Yulan Huang, and Sebastian Scherer. Semantic 3d occupancy mapping through efficient high order crfs. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 590–597. IEEE, 2017. 5

[48] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3947–3956, 2019. 3, 8